



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

| APPLICATION NO. | FILING DATE | FIRST NAMED INVENTOR | ATTORNEY DOCKET NO. | CONFIRMATION NO. |
|-----------------|-------------|----------------------|---------------------|------------------|
| 09/854,031 | 05/11/2001 | Aaron Kershenbaum | YOR920010421US1 | 8153 |

48813 7590 03/22/2005

LAW OFFICE OF IDO TUCHMAN
69-60 108 STREET
SUITE 503
FOREST HILLS, NY 11375

| |
|----------|
| EXAMINER |
|----------|

KIM, JUNG W

| | |
|----------|--------------|
| ART UNIT | PAPER NUMBER |
|----------|--------------|

2132

DATE MAILED: 03/22/2005

Please find below and/or attached an Office communication concerning this application or proceeding.

| | | | |
|------------------------------|-------------------------------|------------------------------------|--|
| Office Action Summary | Application No. 09/854,031 | Applicant(s) KERSHENBAUM ET AL. | |
| | Examiner Jung W Kim | Art Unit 2132 | |

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --
Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

- 1) ☒ Responsive to communication(s) filed on 21 January 2005.
- 2a) ☒ This action is **FINAL**. 2b) ☐ This action is non-final.
- 3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

- 4) ☒ Claim(s) 1-100 is/are pending in the application.
- 4a) Of the above claim(s) _____ is/are withdrawn from consideration.
- 5) ☐ Claim(s) _____ is/are allowed.
- 6) ☒ Claim(s) 1-100 is/are rejected.
- 7) ☐ Claim(s) _____ is/are objected to.
- 8) ☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

- 9) ☐ The specification is objected to by the Examiner.
- 10) ☐ The drawing(s) filed on _____ is/are: a) ☐ accepted or b) ☐ objected to by the Examiner.
Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).
Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

- 12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
a) ☐ All b) ☐ Some * c) ☐ None of:
1. ☐ Certified copies of the priority documents have been received.
2. ☐ Certified copies of the priority documents have been received in Application No. _____.
3. ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).
- * See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

- | | |
|--|---|
| 1) <input type="checkbox"/> Notice of References Cited (PTO-892) | 4) <input type="checkbox"/> Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ |
| 2) <input type="checkbox"/> Notice of Draftsperson's Patent Drawing Review (PTO-948) | 5) <input type="checkbox"/> Notice of Informal Patent Application (PTO-152) |
| 3) <input type="checkbox"/> Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) Paper No(s)/Mail Date _____ | 6) <input type="checkbox"/> Other: _____ |

DETAILED ACTION

1. Claims 1-100 have been examined.

Response to Amendment

2. The 112, second paragraph rejection to claims 50, 60, 61, 71, 81, 87 and 88 for use of terms "useful" and "beneficial" are withdrawn as the amendment overcomes the 112, second paragraph rejections.

Response to Arguments

3. Applicant's arguments filed January 21, 2005 have been fully considered, examiner's response are as follows:
4. Applicant's argument that "C#" and the names of JAVA method and class names is not trademarked, and hence should not be rejected as under 35 U.S.C. 112, MPEP 2173.05(u) and Ex parte Simpson, 219 USPQ 1020 (Bd. App. 1982) has been considered and is persuasive, these rejections are withdrawn. However, applicant's argument against the rejection to the trademark JAVA defined in claims 6, 7, 20, 21, 37, 38, 64-69 is not persuasive since contrary to applicant's argument *that Java, as used in the claims, refers to the generic high-level Java programming language implemented by various organization and companies ... Thus, the term "Java" in the claims does not refer to a specific Java implementation originating from a particular manufacturer*

Art Unit: 2132

(Remarks, pg. 14, 1st paragraph), JAVA was created and trademarked by Sun Microsystems of which the syntactic programming language undergoes constant modification by Sun Microsystems: the JAVA language is divided into two principle packages as of the date of this Office action: J2SE version 5.0 and J2EE version 1.4.2. Hence, the use of the trademark JAVA renders these claims indefinite for failure to adequately define the scope of the claims. MPEP 2173.05(u) ("If the trademark or trade name is used in a claim as a limitation to identify or describe a particular material or product, the claim does not comply with the requirements of the 35 U.S.C. 112, second paragraph. *Ex parte Simpson*, 218 USPQ 1020 (Bd. App. 1982). The claim scope is uncertain since the trademark or trade name cannot be used properly to identify any particular material or product. In fact, the value of a trademark would be lost to the extent that it became descriptive of a product, rather than used as an identification of a source or origin of a product. Thus, the use of a trademark or trade name in a claim to identify or describe a material or product would not only render a claim indefinite, but would also constitute an improper use of the trademark or trade name.")

5. Further, the specific references to the term JAVA as used to define the scope of applicant's claims are used in connection with specific method, class and package names, all of which are enabled in relation to specific versions. Claims that incorporate specific methods, classes and packages by a distributor (in the specific case, the methods, classes and packages listed in and associated with the JAVA 2 platform security architecture), and do not specify the version of such methods render these claims indefinite for failure to adequately define the scope of the claims. MPEP

2173.05(u) (" It is important to recognize that a trademark or trade name is used to identify a source of goods, and not the goods themselves ... If a trademark or trade name appears in a claim and is not intended as a limitation in the claim, the question of why it is in the claim should be addressed. Does its presence in the claim cause confusion as to the scope of the claim? If so, the claim should be rejected under 35 U.S.C. 112, second paragraph.")

6. Regarding applicant's argument that the cited documents do not teach or suggest all the claim limitations (Remarks, pg. 14-24, section I), examiner disagrees. Applicant's arguments are direct against the references individually: the 103 obviousness rejections are made based on the combined teachings and suggestions of Nyanchama in view of Schmidt, Gong and/or Laskoski; however, these arguments are only directed to the Nyanchama reference. One cannot show nonobviousness by attacking references individually where the rejections are based on combinations of references. See *In re Keller*, 642 F.2d 413, 208 USPQ 871 (CCPA 1981); *In re Merck & Co.*, 800 F.2d 1091, 231 USPQ 375 (Fed. Cir. 1986).

7. Regarding applicant's demand to produce documentary evidence of the Official Notice taken for the rejections of claim 4 and 30, it is noted that applicant's traversal is an inadequate challenge to the rejection. A general allegation that "Applicant believes that limitations of claim 4 [and claim 30] are novel in there own right, and therefore cannot agree with the Official Notice assertions in the Office Action" (Remarks, pg. 16,

Art Unit: 2132

3rd paragraph, see also pg. 18, 4th full paragraph) does not establish why the noticed fact is not considered to be common knowledge or well-known in the art. MPEP 2144.03.C. See also 37 CFR 1.111(b). ("specifically point out the supposed errors in the examiner's action, which includes stating why the noticed fact is not considered to be common knowledge or well-known in the art.") Further, as a consequence, the common knowledge or well-known statement in the rejections of claims 4 and 30 are taken to be admitted prior art. MPEP 2144.03.C ("If applicant does not traverse the examiner's assertion of official notice or applicant's traverse is not adequate, the examiner should clearly indicate in the next Office action that the common knowledge or well-known in the art statement is taken to be admitted prior art because applicant either failed to traverse the examiner's assertion of official notice or that the traverse was inadequate")

8. In response to applicant's argument that there is no suggestion to combine the references of Nyanchama and Schmidt (Remarks, pgs 24-25, section II), the examiner recognizes that obviousness can only be established by combining or modifying the teachings of the prior art to produce the claimed invention where there is some teaching, suggestion, or motivation to do so found either in the references themselves or in the knowledge generally available to one of ordinary skill in the art. See *In re Fine*, 837 F.2d 1071, 5 USPQ2d 1596 (Fed. Cir. 1988) and *In re Jones*, 958 F.2d 347, 21 USPQ2d 1941 (Fed. Cir. 1992). In this case, Nyanchama discloses developing role based authorizations to perform tasks with many users and many resources by means of role graphs, wherein the nodes of the graph represent roles in the system.

(Nyanchama, section 2, 'Definitions and Reference Model'; section 3 'Role Graphs'; figure 1) This interpretation is much broader than applicant's interpretation which states that Nyanchama only discloses RBAC algorithms that assist network administrators or security officers to manage access rights for various network users; Nyanchama does not suggest the reference is narrowly defined for only network users and network administrators; rather, Nyanchama maintains the broader teaching. (see Nyanchama, section 1, 'Introduction', section 2.4, 'Authorization') Furthermore, Nyanchama directly ties role graphs with changes in privileges based on inheritance of an object oriented system, which is a unique property found in JAVA object instantiation within a flow of code execution as known to one of ordinary skill in the art; the example of maintaining access to salaries of employees and professors are explained in the context of privileges assigned to objects-"Professors is a subclass of Employees" (see Nyanchama, pg. 7, 2nd full paragraph and figure 1) Schmidt discloses the relationship between a data flow analysis with the programs abstract interpretation. Implicit in Schmidt is the teaching of program code into discrete parts as a means of devising analysis of the code (see Schmidt, Abstract). Such a procurement isolates a program into distinct transitions: the prior art of Nyanchama and Schmidt are relevant subject matters since management of access rights are expressed concerns in any task to access or handle secured objects as taught by Nyanchama; and the transitions defined by Schmidt identifies memory and object access according to data flow (see Schmidt, figure 1). The nexus of the two prior art lies in the well-known fact of JAVA processes: all processes executing JAVA code run as a specific user with privileges and restrictions

Art Unit: 2132

to access libraries, other codes and resources. (see also admitted prior art in applicant's specification "Background of the Invention") Hence, the teaching of translating a program into a graph as a means of analysis, wherein authorization is established by means of a role graph is sufficient motivation to combine the teachings of Nyanchama and Schmidt.

Claim Objections

9. Claim 80 is objected to because of the following informalities: the claim is not grammatical. Appropriate correction is required.

Claim Rejections - 35 USC § 112

10. The following is a quotation of the second paragraph of 35 U.S.C. 112:

The specification shall conclude with one or more claims particularly pointing out and distinctly claiming the subject matter which the applicant regards as his invention.

11. Claims 6, 7, 10, 20, 21, 24, 37, 38, 41, 64, 65-69 and 73-80 are rejected under 35 U.S.C. 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention.

12. As per claims 6, 7, 10, 20, 21, 24, 37, 38, 41, 64, 65-69 and 73-80, the presence of the trademarks or trade names "JAVA" are not proper under 35 U.S.C. 112, second paragraph (see 37 CFR 2173.05(u)).

13. If trademarks or trade names, or names of a method, class or package are used in a claim as a limitation to identify or describe a particular material or product, the claim

does not comply with the requirements of the 35 U.S.C. 112, second paragraph. Ex parte Simpson, 218 USPQ 1020 (Bd. App. 1982). The scope of the claims is uncertain since a trademark or trade name cannot be used properly to identify any particular material or product.

Claim Rejections - 35 USC § 103

14. The text of those sections of Title 35, U.S. Code not included in this action can be found in a prior Office action.

15. Claims 1-5, 11-19, 25-36, 42-50, 59-62, 70-72 and 82-100 are rejected under 35 U.S.C. 103(a) as being unpatentable over Nyanchama "The Role Graph Model and Conflict of Interest" (hereinafter Nyanchama) in view of Schmidt "Data Flow Analysis is Model Checking of Abstract Interpretations" (hereinafter Schmidt).

16. As per claim 1, Nyanchama teaches a method comprising employing a computer for:

- a. obtaining a system defined by a set of authorizations;
- b. providing a graph representing the system defined by a set of authorizations;
- c. identifying any authorization resources associated in the graph as nodes;
- d. locating any bounded path within the graph; and
- e. associating the any authorization resources with the any bounded path.

See Nyanchama, pages 10-15, section 4, 'Role Graph Administration Algorithms'; Figure 2; Table 1; pages 5-9, section 2, definitions and reference model, 'Privileges', 'Roles', 'Authorization', 'Policies'.

17. Nyanchama does not expressly teach translating a collection of code into a graph for analysis. Schmidt teaches translating a program into a graph to perform model checking. See Schmidt, pages 39-41, section 3, 'Trace-Based Abstract Interpretation'; pages 41-42, 'Collecting Semantics'; pages 43-44, section 6, 'Why a Data-Flow Analysis is a Model Check'. It would be obvious to one of ordinary skill in the art at the time the invention was made to translate a program into a graph to analyze the state of the program as known to one of ordinary skill in the art and as taught by Schmidt. Ibid. The aforementioned cover the limitations of claim 1.

18. As per claim 2, Nyanchama covers a method as outlined above in the claim 1 rejection under 35 U.S.C. 103(a). In addition, the collection of code includes codes obtained from a group of codes including basic blocks, class methods, classes, collections of classes or any combination of these. See Nyanchama, page 5, 2nd full paragraph and 4th full paragraph; see Schmidt, page 39, last paragraph; page 41, section 4, 'Collecting Semantics' and Figure 2. The aforementioned cover the limitations of claim 2.

19. As per claim 3, Nyanchama covers a method as outlined above in the claim 1 rejection under 35 U.S.C. 103(a). In addition, the step of providing includes

constructing the program graph through static analysis techniques (abstract interpretations). See Schmidt, page 39, section 2, last paragraph. The aforementioned cover the limitations of claim 3.

20. As per claim 4, Nyanchama covers a method as outlined above in the claim 3 rejection under 35 U.S.C. 103(a). In addition, employing object code or any intermediary state of a program is the standard means of constructing graphs to analyze the model of a program. For example, compiler programs translate source code into object code to perform optimizations on the code. Examiner takes Official Notice of this teaching. It would be obvious to one of ordinary skill in the art at the time the invention was made to employ object code to construct the program graph so that analysis of the program will be based on object code rather than source code, which is geared to human-readability. The aforementioned covers the limitations of claim 4.

21. As per claim 5, Nyanchama covers a method as outlined above in the claim 1 rejection under 35 U.S.C. 103(a). In addition, the step of identifying includes finding at least one authorization point in the program graph. See Nyanchama, page 11, Table 1, 'direct privileges' and 'effective privileges'. The aforementioned cover the limitations of claim 5.

22. As per claim 11, Nyanchama covers a method as outlined above in the claim 1 rejection under 35 U.S.C. 103(a). In addition, the step of identifying includes employing

data flow analysis. See Schmidt, page 39, section 2, last paragraph, 2nd sentence. The aforementioned cover the limitations of claim 11.

23. As per claim 12, Nyanchama covers a method as outlined above in the claim 11 rejection under 35 U.S.C. 103(a). In addition, the step of employing includes generating a data flow from the program graph. See Schmidt, page 40, Figure 1. The aforementioned cover the limitations of claim 12.

24. As per claim 13, Nyanchama covers a method as outlined above in the claim 1 rejection under 35 U.S.C. 103(a). In addition, the step of identifying any bounded path includes locating a set of start nodes in the program graph, and locating a stop node in the program graph; and the bounded path includes all nodes within the graph bound by the start nodes and the stop node. See Schmidt, page 40, Figure 1, 'Concrete computation tree'. The aforementioned cover the limitations of claim 13.

25. As per claim 14, Nyanchama covers a method as outlined above in the claim 1 rejection under 35 U.S.C. 103(a). In addition, the step of associating includes associating and aggregating the any authorization resource with the collection of code. See Nyanchama, page 5, section 2.1, 'Privileges'; page 9, section 3, 'Role Graphs'; page 11, Figure 2. The aforementioned cover the limitations of claim 14.

26. As per claims 15-19 and 25-28, they are claims corresponding to claims 1-5 and 11-14, and they do not teach or define above the information claimed in claims 1-5 and 11-14. Therefore, claims 15-19 and 25-28 are rejected as being unpatentable over Nyanchama in view of Schmidt for the same reasons set forth in the rejections of claims 1-5 and 11-14.

27. As per claims 29 and 31, they are claims covered by the inventions outlined in the claim 1-5 and 11-14 rejections, and they do not teach or define above the information in the claim 1-5 and 11-14 rejections. Therefore, claims 29 and 31 are rejected as being unpatentable over Nyanchama in view of Schmidt for the same reasons set forth in the rejections of claims 1-5 and 11-14.

28. As per claim 30, Nyanchama covers a method as outlined above in the claim 29 rejection under 35 U.S.C. 103(a). In addition, a step that provides an indication that operations dependent on a property are not necessary when the property has not been identified or is not identified is a standard coding practice. This step prevents superfluous operations. See MPEP 2144.04.II.A 'Elimination of a step or an element and it's function'. Examiner takes Official Notice of this teaching. It would be obvious to one of ordinary skill in the art at the time the invention was made to provide an indication that authorization testing is not necessary when no resource is identified by the method to make for a more efficient method as known to one of ordinary skill in the art. The aforementioned cover the limitations of claim 30.

29. As per claims 32-36, 42 and 43, Nyanchama covers an apparatus as outlined above in the claim 15-19 and 25-28 rejections under 35 U.S.C. 103(a). In addition, a means to identify any authorization resources within the collection of code is an authorization resource identifier; a means to locate any bounded path within a program graph of the collection of code is a bounded path locator; a means to associate any authorization resource with the any bounded path is an associator; and a means to construct the program graph is a program graph constructor. The aforementioned cover the limitations of claims 32-36, 42 and 43.

30. As per claims 44-46, they are claims corresponding to claims 29-36, 42 and 43, and they do not teach or define above the information claimed in claims 29-36, 42 and 43. Therefore, claims 44-46 are rejected as being unpatentable over Nyanchama in view of Schmidt for the same reasons set forth in the rejections of claims 29-36, 42 and 43.

31. As per claims 47-49, they are claims corresponding to claims 29-36, 42 and 43, and they do not teach or define above the information claimed in claims 29-36, 42 and 43. Therefore, claims 47-49 are rejected as being unpatentable over Nyanchama in view of Schmidt for the same reasons set forth in the rejections of claims 29-36, 42 and 43.

32. As per claims 50, 59-62 and 70-72, they are claims corresponding to the inventions outlined in the claim 1-5 and 11-14 rejections, and they do not teach or define above the information outlined in the claim 1-5 and 11-14 rejections. Therefore, claims 50, 59-62 and 70-72 are rejected as being unpatentable over Nyanchama in view of Schmidt for the same reasons set forth in the rejections of claims 1-5 and 11-14.

33. As per claims 82-91, they are claims corresponding to the inventions covered by the claim rejections as listed above, and they do not teach or define above the information outlined. Therefore, for the reasons listed above, claims 82-91 are rejected as being unpatentable over Nyanchama in view of Schmidt.

34. As per claims 92-100, they are article of manufacture claims, computer program product claims, and program storage device claims corresponding to the inventions outlined in the claim 1-5, 11-19, 25-36, 42-50, 59-62, 70-72 and 82-91 rejections, and they do not teach or define above the information outlined in the claim 1-5, 11-19, 25-36, 42-50, 59-62, 70-72 and 82-91 rejections. Therefore, claims 92-100 are rejected as being unpatentable over Nyanchama in view of Schmidt for the same reasons set forth in the rejections of claims 1-5, 11-19, 25-36, 42-50, 59-62, 70-72 and 82-91.

35. Claims 6-10, 20-24, 37-41, 52-58, 63-69 and 73-81 are rejected under 35 U.S.C. 103(a) as being unpatentable over Nyanchama in view of Schmidt, and further in view of Gong "Java Security Architecture (JDK 1.2)" (hereinafter Gong).

36. As per claims 6-9, Nyanchama covers a method as outlined above. Nyanchama does not expressly teach using JAVA and the JAVA Security Architecture to determine authorization points. Gong discloses a package to check for access authorization of a code using an AccessController object. See Gong, pages 31-33, section 4.2, `java.security.AccessController`. Further, methods that instantiate an AccessController object and call the `checkPermission()` method are finding authorization points. Moreover, the function call to invoke the AccessController methods is an instruction invocation. Therefore, it would be obvious to one of ordinary skill in the art at the time the invention was made for the program to be a JAVA program and for the steps to find an authorization point be implemented using the JAVA Security Architecture since JAVA has become a widely used language to create OO programs and Sun Microsystems has provided the JAVA Security Architecture to secure programs written in the JAVA language. See Gong, page 1, Introduction. The aforementioned cover the limitations of claims 6-9.

37. As per claim 10, Nyanchama covers a method as outlined above in the claim 6-9 rejections under 35 U.S.C. 103(a). In addition, C# is another popular OO programming language provided by MICROSOFT. Hence, it would be obvious to one of ordinary skill in the art at the time the invention was made for the particular language to be C#, since C# offers many of the modularity benefits of the JAVA language as known to one of ordinary skill in the art. The aforementioned cover the limitations of claim 10.

38. As per claims 20-24, they are claims corresponding to claims 6-10, and they do not teach or define above the information claimed in claims 6-10. Therefore, claims 20-24 are rejected as being unpatentable over Nyanchama in view of Schmidt and Gong for the same reasons set forth in the rejections of claims 6-10.

39. As per claims 37-41, they are claims corresponding to claims 20-24, 32-36, 42 and 43, and they do not teach or define above the information claimed in claims 20-24, 32-36, 42 and 43. Therefore, claims 37-41 are rejected as being unpatentable over Nyanchama in view of Schmidt and Gong for the same reasons set forth in the rejections of claims 20-24, 32-36, 42 and 43.

40. As per claims 52 and 53, Nyanchama covers a method as outlined above in the claim 6-10 and 50 rejections under 35 U.S.C. 103(a). In addition, the step of constructing includes the step of building an invocation graph and a call graph of the collection of code to form the program graph. See Schmidt, page 40, Figure 1 and page 41, Figure 2; see Gong, pages 31-32, 3 bullets. The aforementioned cover the limitations of claims 52 and 53.

41. As per claims 54-58, Nyanchama covers a method as outlined above in the claim 52 and 53 rejections under 35 U.S.C. 103(a). In addition, the JAVA Security Architecture enables authorization identification using context-sensitivity (see Gong,

page 37, section 4.3, 'Inheritance of Access Control Context'), wherein context sensitivity includes using type information for any method receiver and/or any parameter (see Gong, pages 30-31, section 4.1: a class belongs to one protection domain); wherein the step of using type information includes using class and memory allocation site information which includes using per instance information (objects passed by reference as a method parameter are instantiated classes in JAVA). Furthermore, this instance information is associated with a node or edge in the program graph. See Nyanchama, page 11, Figure 2: only allocated resources have a definite authorization level. The aforementioned cover the limitations of claims 54-58.

42. As per claims 63 and 64, they are claims corresponding to claims 6-10 and 50, and they do not teach or define above the information claimed in claims 6-10 and 50. Therefore, claims 63 and 64 are rejected as being unpatentable over Nyanchama in view of Schmidt and Gong for the same reasons set forth in the rejections of claims 6-10 and 50.

43. As per claims 65-69 and 73-81, Nyanchama covers a method as outlined above. In addition, the JAVA Security Architecture enables a resource identifier to include at least one `java.security.Permission` object (see Gong, pages 8-9, sections 3.1 and 3.1.1; page 39, '`acc.checkPermission(permission)`'); wherein the authorization test is a call to any `java.security.AccessController.checkPermission()` method (see Gong, page 31-32, section 4.2, three bullets); wherein the location represents a call to any authorization

testing method in any instance of `java.lang.SecurityManager` and/or one of its subclasses (see Gong, page 45, section 6.2 and by the property of JAVA class inheritance); wherein the node has a parameter which the type information is a JAVA `java.security.Permission` (see Gong, page 39, '`acc.checkPermission(permission)`'); wherein the step of identifying includes locating the constructor for the JAVA `java.security.Permission` allocation site and using the data flow analysis in identifying any value passed by any parameter to the constructor, wherein the combination of the JAVA `java.security.Permission` and a value for any parameter is the used Permission (see Gong, page 33-34, section 4.2.1, 'Algorithm for Checking Permissions'; pages 38-40, section 4.4; see Schmidt, pages 41-42, section 4, 'Collecting Semantics'); wherein the method further comprising employing a privileged JAVA code wherein the stop node represents the method `java.security.AccessController.checkPermission()`, and employing the start nodes are any of the root nodes in the program graph or a node representing the method `java.security.AccessController.doPrivileged()`, and connecting a used Permission with the privileged JAVA code (see Gong, pages 33-34, section 4.2.1 'Algorithm for Checking Permissions', especially page 33, last paragraph; page 35, 3rd paragraph, 'normal use of the "privileged" feature'); wherein the step of associating includes connecting a used Permission with any node in the program graph prior to the `java.security.AccessController.doPrivileged()` node (see Gong, pages 33-34, section 4.2.1 'Algorithm for Checking Permissions': a caller whose domain is granted the permission must be marked as "privileged"); wherein the step of associating includes connecting a used Permission for each

java.security.AccessController.checkPermission() in the program graph (Permission object is a necessary parameter to call the checkPermission method); wherein the step of associating includes connecting the used Permission from each node in the program graph to each method and from each method to each class and from each class to a collection of classes (see Schmidt, page 11, Figure 2; see Gong, page 6, Figure; pages 8-9, section 3.1-3.1.3); and wherein the method further comprising employing the resource in executing the collection of code (see Gong, page 35, 'somemethod()'). The aforementioned cover the limitations of claims 65-69 and 73-81.

44. Claim 51 is rejected under 35 U.S.C. 103(a) as being unpatentable over Nyanchama in view of Schmidt, and further in view of Laskoski U.S. Patent No. 5,428,554 (hereinafter Laskoski).

45. As per claim 51, Nyanchama covers a method as outlined above in the claim 50 rejection under 35 U.S.C. 103(a). Nyanchama does not expressly disclose the step of constructing including employing source code of the collection of code. Laskoski teaches employing source code to form a directed graph. See Laskoski, col. 2, lines 30-35. It would be obvious to one of ordinary skill in the art at the time the invention was made to employ source code of the collection of code to construct a program graph to improve a programmer's comprehension of program resource allocation within the collection of code. See Laskoski, col. 1, lines 1-5. The aforementioned cover the limitations of claim 51.

Conclusion

46. **THIS ACTION IS MADE FINAL.** Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire **THREE MONTHS** from the mailing date of this action. In the event a first reply is filed within **TWO MONTHS** of the mailing date of this final action and the advisory action is not mailed until after the end of the **THREE-MONTH** shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than **SIX MONTHS** from the mailing date of this final action.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Jung W Kim whose telephone number is (571) 272-3804. The examiner can normally be reached on M-F 9:00-5:00.

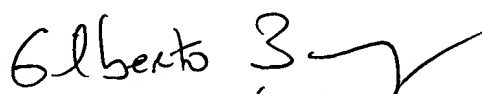
If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Gilberto Barron can be reached on (571) 272-3799. The fax phone number for the organization where this application or proceeding is assigned is 703-872-9306.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).



Jung W Kim
Examiner
Art Unit 2132

Jk
March 15, 2005



GILBERTO BARRON JR.
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100